State of Michigan
Road Quality Forecasting System
Test Plan

General Information

Project ID / Acronym:	RQFS	***Date:***	10/02/2007
Organizational Title:	MDOT	***Modification Date:***	10/05/2008
Author(s):	Bob Baker - PM	***MDIT Authorized by:***	Viji Jayaraman

Privacy Information

This document may contain information of a sensitive nature. This information should not be given to persons other than those who are involved in the Road Quality Forecasting System project or who will become involved during the lifecycle.

Change Control

The following information is being used to control and track modifications made to this document.

Revision Date	Author	Section(s)	Summary
10/05/2007	**Bob Baker**		**Added updates from walkthrough**

This example is from MiCSES which is a large and complex project. This project is a Federally Mandated System with approximately 200+ staff.

Table of Contents

1. Introduction

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope..

The RQFS project is being developed in two phases. The first phase will have the strategy management functionality which is the main function of the RQFS application.

System Testing, User Acceptance Testing, Quality Assurance Testing and Load /Stress Testing will occur prior to the conclusion of each Phase.

This test plan may be further updated, to accurately reflect testing needs that may be discovered as the team progresses from phase one to phase two.

1.1 Objectives

Usability Testing

Users of the application are invited to perform common functions and provide feedback on the usability of the new application.

- Is the new application easy to navigate?
- Does the page flow match workflow?
- Does the application meet the user's needs?

Unit Testing

Developers conduct this level of testing during construction. The main focus of Unit Testing is to:

- Validate proper function of design changes/customizations
- Validate field level edits and validation

Integration Testing

Developers are responsible for ensuring the individual modules are combined and tested as a group. This type of testing follows unit testing and precedes system testing. Integration Testing can be used to:

- Validate proper function of all Integration Points
- Validate all application functionality (business logic) through test scripts
- "Try to break it" during abnormal use

Standards Review Testing

The testing staff (QA Team) will perform this testing at the conclusion of each iteration cycle and again during Systems Testing before each Phase Release. Only the components that have been completed to date will be tested. The goals are to:

- Ensure State of Michigan and MDOT standards are met
- Increase ease-of-use, reliability and performance by adhering to developed standards, practices and procedures used at MDOT and the State of Michigan.

System Testing

The testing staff (QA Team) will perform this testing which attempts to discover defects by doing an end-to-end test of the business processes within the application rather than of its individual components. Further definition of what System Testing covers is:

- Validate end-to-end system functionality prior to User Acceptance Testing (UAT)

- Exploratory Testing
- Test the function of an application as a whole (or as much as the team has completed to date)

Load Testing

The testing team (QA Team) will perform load testing on applications. There are many different types of metrics that can be measured. The testing resource can help each project team determine what types of metrics best suit the needs of the individual project. Load Testing is used to:

- Validate the system performance in a high transaction rate or high amount of users
- Ensure the environment stability

Regression Testing

Three FTEs will perform regression testing by re-running test cases from prior iterations to ensure that the introduction of new functionality did not impact functionality from a prior iteration.

User Acceptance Testing (UAT)

Business owners and / or selected end users can be employed to perform UAT. UAT is the final testing phase and occurs after Integration and System testing have been completed and prior to production release. The goal of UAT is to:

- Validate proper functionality of the application by the stakeholders (business users) based upon original requirements of the project.

The end result of these tests is to indicate whether the system is ready to be used in a production environment or not (GO, NO-GO).

1.2 Scope

The QC Team will perform Standard, System and Load Testing for completed items in RQFS, providing feedback to the development team in the form of updateable issue tracking logs and reports. The QA team will re-test any issues they reported to confirm the corrections have been completed and update the logs appropriately. The goal is to have any reported issues in an acceptable state for the production roll-out of the first phase of RQFS. (No “new” or “open” issues. Logged issues should be in a status of “closed”, N/A” or “pending” prior to release.)

Due to the 2-phase roll-out approach of this project, not all functionality will be available until the last release. The application should inform the user of any items that are in a non-active status during a given release.

The existing application, RQFS, will continue to function. Comparison testing between the two applications will be of little value due to changes in the calculations and additional options available with the new application.

1.3 Responsibilities

Development Team: Development team members will create the RQFS Test Cases to help confirm and identify business requirements for the web items completed for each phase. The team will also perform unit and functionality testing as they develop the code. The Development Team includes:

Bhushan Ramning

Bharath Krishnan

Quality Assurance Team: The QA Team will conduct Standards Review, System Testing, and Load/Stress Testing. The members of the QA Team and their involvement include:

Jan Miller – Standards Review and System Testing

Sathya Murthy and / or James Anthony – Load / Stress Testing

Business Client: The business client will use the Test Cases to confirm that the components meet the business needs. Each test case will be marked as passed or failed in a tracking log, so that any issues or changes can be identified and tracked. The business client responsible for these activities is:

Jon Galbreath

Ben Krom

2. Project Details

2.1 Test Environment

The QA Server is MDOT8.

The UARS welcome after sign-in will provide a hyperlink to the RQFS:
https://wastest.mdot.state.mi.us/loginqa/action/uarsWelcome.do

2.2 Important Functionalities (Modules) to be tested

Module No.	Functionalities/Features	Suitable Testing Types
1	Security Access "Handshake" between RQFS and UARS	Standards Testing, System Testing, Review
2	Preservation Strategy – Process, Create, Update, Inquire, View, Delete	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
3	Fix strategy – Add, Update, Delete, Insert	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
4	Create Strategy from Projects	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
5	Combine multiple strategies	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
6	Print Strategies	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
7	Create a spreadsheet	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
8	Generate graphs	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
9	Convert from Percentage to Lane Miles and visa versa.	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
10	Get RSL data from PaveMapp	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
11	Generate Strategy analysis results by year or Catagory	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing

Module No.	Functionalities/Features	Suitable Testing Types
12	Ability to select mutually exclusive networks	Unit Testing, Integration Testing, Standards Testing, System Testing, Load Testing, User Acceptance Testing, Regression Testing
13	Role Base Security	System Testing, User Acceptance Testing

2.3 Performance Testing Procedures

DIT will use Mercury Load Runner to test for heavy load scenarios. DIT staff will work closely with the development team to ensure the needs of RQFS are met. Load/Stress Testing will occur prior to each roll-out Phase release (two times.)

2.4 Stress Testing Procedures

DIT will use Mercury Load Runner to test for heavy load scenarios. DIT staff will work closely with Infrastructure staff and the development team to analyze what the behavior is when a high load scenario happens to ensure the user is able to recover from any errors without PC or Server corruption.

Load/Stress Testing will occur prior to each roll-out Phase release (two times.)

2.5 Application Testing Procedures

Final testing, including user testing, and defect rework will be done in the QA environment which simulates the production environment.

2.6 Interface Testing Procedures

Use Test Cases will be used by the clients, business owners and additional testers during Integration/System Testing, User Acceptance Testing and Regression Testing. The QA Team will perform Standards Review Testing and will conduct System Testing and Comparison Testing prior to each Phase Release. Each input field will be tested for values in-bounds, at bounds and out-of-bounds. The value type, length and interdependency between fields will be tested. The application will be tested for both correct and incorrect input, and the messages and exception handling of the application will also be tested.

Interfaces include MAP and PaveMapp.

The new RQFS will interface with the PaveMapp system when the “Create Strategy” page is loaded. This interface will also be activated for “View a Strategy” and “Combined a Strategy” functionality. This interface will retrieve the “Remaining Service Life” (RSL) of the section of road being included in the strategy along with the network lane miles. During the “Process Strategy” functionality RQFS will access RSL data from PaveMapp

This interface can be tested successfully if correct lane miles are displayed on strategy definition page or check if application gives message "RSL data not found" if data for selected network is not present in PaveMapp.

RQFS accesses MAP & PaveMapp for “Program Projects functionality”. RQFS will read project segments from MAP for given network combination. To successfully test this use case, application should retrieve proper data for given network combination.

2.7 System Recovery Testing Procedures

DIT will use Mercury Load Runner to test for heavy load scenarios. DIT staff will work closely with Infrastructure staff and the development team to analyze what the behavior is when a high load scenario happens to ensure the user is able to recover from any errors without PC or Server corruption.

2.8 Features Outside of Scope

All phase two functionality is out of scope for the phase one testing.

3. Test Execution

3.1 Test Schedule

Activity	Start Date	End Date	Responsibility
Unit Testing		10/05/2007	Dev. Team
Integration Testing	10/08/2007	11/02/2007	Dev. Team
Move RQFS from Dev to QA	11/05/2007	11/07/2007	QA Team
Standards, System and Load Testing	11/07/2007	12/14/2007	QA Team
Regression Testing	11/26/2007	12/14/2007	Dev. Team
User Acceptance Testing	11/26/2007	12/14/2007	Business Client

3.2 Test Progression

Module No.	Functionalities/Features
1	Security Access “Handshake” between RQFS and UARS
2	Create Strategy
3	Inquire Strategy
4	Combine Strategy

For each phase, the following types of testing are performed: Unit, Integration, Standards Review, System, Load, Regression if applicable, and User Testing

Any issues found during testing are reported to the Project Manager and documented in the defect log. Any fix must be re-tested and approved by the same person who reported the issue in order for the issue to be marked as “closed”.

3.3 Monitoring

Issue Tracking logs will be utilized to record and track the status of any suspected issues. Prior to the go-live production date, all issues should be in a “non-open” state. (i.e. “Closed”, “N/A” or “Pending”.) Any open bugs should be reported to the development team’s manager for final “Go / No Go” decision.

Team will meet during the testing phase to determine progress and discuss any issues and to plan rework needed based on findings from the testing.

3.4 Reporting

Comparison Test Cases will show what was tested and passed/failed.

Use Test Cases created by developers will show what the business owner / selected end users tested and passed/failed.

Defects will be captured in a defect log and communicated back to the appropriate resource at the conclusion of that testing iteration. During System Testing, any high priority defects will be immediately reported to the Project Manager for evaluation and fix.

4. Attachments

Attach specific test reports.

Approval Information

The signatures relay an understanding of the purpose and content of the document by those endorsing it.

☐ Approve ☐ Approve with Modifications ☐ Reject

Comments:

"Click HERE and Type"

	Name / Title	Initial Signoff	Date	Final Signoff	Date
Client Sponsor					
DIT Sponsor					
Project Manager					